METHOD AND SYSTEM FOR INCREASING DATA ACCESS IN A SECURE SOCKET LAYER NETWORK ENVIRONMENT

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to the transfer of data over a Secure Socket Layer (SSL). More particularly, the invention is directed to a method and system for increasing data access in a secure socket layer network environment.

Description of the Prior Art

The Secure Socket Layer (SSL) protocol (see The SSL Protocol Version 3.0, http://www.netscape.com/eng/ssl3/draft302.txt) is presently the de facto industry standard for Web security. It is common for E-commerce applications which are based on the Web to employ the SSL protocol which is built into almost all Web servers and browsers, such as Netscape Enterprise Server, Microsoft Web Server, Netscape Communicator, and Microsoft Internet Explorer (IE).

The SSL protocol uses public key cryptography in conjunction with an X.509 certificate to provide server authentication and, optionally, client authentication. During the authentication process, the server sends its X.509 certificate chain which may or may not contain the root CA certificate to the client as part of the handshake messages the client and server exchange at the start of a session. The client validates the server's certificate through a normal certificate verification procedure if it has the server's certificate, it has the certification authority's (CA) certificate that signed the server's certificate, and it has associated trust information.

While SSL protocol provides for a secure way to transfer data, it currently restricts the

way in which data can be transmitted. Currently, once the client and server establish a connection over SSL, it precludes other technology from acting on the data, e.g., terminating the SSL connection, further processing the de-encrypted data and then compressing the data, since the CA certificate from which the original SSL session was established with would be lost in so doing.

Accordingly, the data is unable to be accelerated through advanced compression technology. Rather, the data must be passed along without intervention. It is paramount to preserve the integrity of SSL protocol meaning that a private key on the server never be passed over the connection, rather only the public key be transmitted.

There remains therefore a need in the industry to accelerate data transfer using the SSL protocol. There also is a need to present the client with a proper CA certificate in order to assure a validated SSL protocol when performing acceleration or increasing data access over a SSL network environment.

BRIEF SUMMARY OF THE INVENTION

It is an object to accelerate data transfer using SSL protocol.

It is another object to maintain a proper CA certificate in order to assure verified SSL protocol when performing acceleration or increasing data access over a SSL network environment.

It is another object to securely project the CA certificate over a SSL connection while employing data acceleration technology.

It is an object to provide a system for increasing data transfer over a SSL network environment.

Accordingly, the present invention is directed to a method and system for increasing data access in a secure socket layer network environment. The system includes a web server computer having SSL protocol server software operably associated therewith for enabling a SSL connection, wherein SSL protocol server software includes a CA certificate and private key, a client computer communicatively linked to the web server computer and has web browser software which includes SSL protocol client software operably associated with the web server for enabling an SSL connection between the client and the web server, and means to verify CA certificate.

SSL acceleration client software is operably disposed on the client computer for monitoring when the web browser requests a SSL connection with the web server. The SSL acceleration client software intercepts the SSL request from the web browser and uses an established SSL connection or initiates another SSL request to the SSL acceleration server computer to establish a SSL connection.

SSL acceleration server software is operably associated with the web server computer for communicating with the SSL acceleration client software and also monitoring when the web server computer receives a request for a SSL connection through SSL acceleration client software. The SSL acceleration server software is operably associated with the SSL protocol server software to obtain a pseudo CA certificate and access to the private key.

Upon detection of the SSL connection request from the web browser software by the SSL acceleration client software, this initiates a SSL handshake between the SSL acceleration client and the SSL acceleration server wherein the pseudo CA certificate is sent to the SSL acceleration client software which includes a public key. The SSL acceleration client then presents the pseudo

CA certificate to the web browser for validation.

Web browser software sends a list of available encryption algorithms (ciphers) back through SSL acceleration client software. The SSL acceleration client software sends a chosen cipher to the browser software. The browser software creates a secret key, encrypts this using chosen cipher and using the previously received public key and sends the encrypted secret key to the SSL acceleration server software through the SSL acceleration client, where SSL acceleration server software de-encrypts the secret key using the copy of the private key and then returns the de-encrypted secret key back to the SSL acceleration client over the existing SSL session between SSL acceleration client and SSL acceleration server. The SSL acceleration client then uses the de-encrypted secret key to complete the SSL handshake between the SSL acceleration client and the web browser software. Once the "handshake" is completed and secure communications between the client computer's web browser and SSL acceleration client software can take place which in turn can communicate with the SSL acceleration server employing advanced acceleration techniques.

Other objects and advantages will be readily apparent to those skilled in the art upon viewing the drawings and reading the detailed description hereafter.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1A is an illustration of the state of the art for SSL data transfer.

FIG. 1B is a flow chart showing the state of the art.

FIG. 2A is an illustration of a SSL data transfer using the present invention.

FIG. 2B is a flow chart employing the invention.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to the drawings, FIGS.1A and 1B depict and the current state of the art. FIG. 1A depicts a web server computer 10 and a client computer 12. Conventionally, a "handshake" connection using SSL is performed as follows. The web server computer 10 is understood to include an operating system/software, server software, memory and linking devices as is known in the art. The web server computer 10 also has SSL protocol server software operably disposed thereon for enabling a SSL connection, wherein SSL protocol server software includes a CA certificate and private key for a particular owner/issuer. The client computer 12 likewise includes an operating system/software, web browser software having SSL protocol client software operably disposed thereon for enabling a SSL connection, memory and linking devices as is known in the art and is communicatively linked to the web server computer 10.

In this way, the steps shown in FIG. 1B can be performed. The client computer 12 initiates 100 a "handshake" by requesting secure connection with the server computer 10. The server computer 10 sends 102 server's CA certificate to client computer 12 along with a public key.

The web browser validates 104 the CA certificate. Web browser sends 106 available ciphers to server computer 10. The server computer 10 sends 108 chosen cipher to client computer 12. Web browser creates 110 a secret key and encrypts using server's public key. The client computer 12 sends 112 the encrypted secret key to the server computer 10. Server computer 10 de-encrypts 114 the secret key using server's private key to establish a handshake. Thus, a secure communication link is established through which communication can take place.

The present invention is generally depicted in FIGS. 2A and 2B and is directed to a system and method for increasing data access in a secure socket layer network environment and is generally designated by the number 100. The system 100 includes a web server computer 102 which has an operating system/software, server software, memory and linking devices as is known in the art. Further, the computer 102 has SSL protocol server software operably disposed thereon for enabling a SSL connection, wherein SSL protocol server software includes a CA certificate and private key.

A client computer 104 includes an operating system/software, web browser software having SSL protocol client software operably disposed thereon for enabling a SSL connection, memory and linking devices as is known in the art and is communicatively linked to the web server computer 102. SSL acceleration client (SSLAC) software is operably disposed on the client computer 104 for monitoring when the web browser requests a SSL connection with the web server 102.

SSL acceleration server (SSLAS) software is operably disposed on the web server computer 104 for receiving a request for a SSL connection through SSL acceleration client software. The SSL acceleration server software is operably associated with the SSL protocol server software to obtain one either a copy or an equal credential of the CA certificate (i.e., a pseudo CA certificate) and private key.

The operation of the invention can be understood from steps shown in FIGS. 2A and 2B. SSL acceleration client software intercepts 200 new SSL request for a SSL secure connection from the web browser to a target web server. The SSL acceleration client software then initiates 202 a SSL handshake with the SSLAS operably associated with the target web server computer

and to start SSL connection. The SSLAS then determines 204 which CA certificate is operably assocated with the target web server. As part of the SSL handshake between SSLAC and SSLAS, the SSLAS sends 206 this CA certificate to SSLAC along with a public key. At this point a secure SSL session is established between SSLAC and SSLAS and all subsequent data traffic between SSLAC and SSLAS flows over this secure connection. The SSLAC software sends 208 the copy of the CA certificate to the web browser for validation 210. Web browser software sends 212 a list of available encryption algorithms (ciphers) back to target web server (i.e., server computer 102). SSLAC software intercepts this from the browser and sends 214 a chosen cipher to the browser software. The web browser software creates 216 a secret key, encrypts using chosen cipher and using the previously received public key and sends 218 the encrypted secret key to the target server, which is intercepted and sent 219 through the SSL acceleration client software to the SSLAS software. SSLAS software de-encrypts 220 the secret key using the private key operably associated with the target server. SSLAS software sends 222 decrypted secret key back to SSLAS software via the secure SSL connection, wherein a "handshake" is completed and secure communications between the client computer's web browser and SSLAS software and by using the secret key, data can be accelerated between the client computer 104 and the web server computer 102 employing acceleration software, such as compression software of the SSL acceleration client/server software.

Because the SSL connection is terminated by SSLAC, SSLAC can process the data in unencrypted form allowing it to apply data compression and other optimization techniques to the data stream. This is done in such a way that the credentials of the SSLAS are presented to the web browser without having violated the SSL paradigm because the private key of the SSLAS

was never transmitted to SSLAC.

The above described embodiment is set forth by way of example and is not for the purpose of limiting the present invention. It will be readily apparent to those skilled in the art that obvious modifications, derivations and variations can be made to the embodiments without departing from the scope of the invention. For example, SSLAS does not need to reside on the web server, but it is contemplated that SSLAS will be remotely located on another computer interacting with the web server computer; or where SSLAC is running on a different computer than browser and can whose services can be shared concurrently by multiple browsers. Accordingly, the claims appended hereto should be read in their full scope including any such modifications, derivations and variations.

What is claimed is: